CLAIMS

1. A method for encoding and decoding first and second data streams comprising:

encoding said first data stream using a first encoder to produce a first encoded data stream;

encoding said second data stream using a second encoder to produce a second encoded data stream;

providing said first and second encoded data streams to a receiver;

decoding said first and second encoded data streams using a single decoder.

2. The method of claim 1 wherein said encoding and decoding are lossless.

3. The method of claim 1 wherein said encoding and decoding are near-lossless.

4. The method of claim 1 wherein said receiver is provided one of said first and second data streams as side-information.

5. The method of claim 4 wherein encoding of said second stream satisfies a prefix condition and said prefix condition is satisfied for a code γ_Y for Y given X when for each $x \in \mathcal{X}$, and each $y, y' \in A_x$, the description of y is not a prefix of the description of y'.

6. The method of claim 5 wherein said code γ_Y is a matched code.

7. The method of claim 6 wherein said code γ_Y is an instantaneous, side-information matched code for $p(x,y)$ when γ_Y is a matched code for some partition $\mathcal{P}(\mathcal{Y})$ for $p(x,y)$.

8. A method of generating code comprising:

obtaining an alphabet of symbols generated by a data source;

identifying combinable symbols of said alphabet and generating subsets of combinable symbols;

identifying optimal partitions of said subsets of symbols to generate a list of groups;

using said list of groups to generate partitions of the full alphabet.

9. The method of claim 8 further comprising determining a matched code for each partition.

10. The method of claim 8 further comprising selecting a partition whose matched code has a best rate.

11. The method of claim 8 wherein said matched code comprises a Huffman code.

12. The method of claim 8 wherein said matched code comprises an arithmetic code.

13. The method of claim 8 wherein symbols $y_1, y_2 \in \mathcal{Y}$ can be combined under $p(x,y)$ if $p(x,y_1)p(x,y_2)=0$ for each $x \in \mathcal{X}$.

14. The method of claim 13 wherein for each symbol a set $\mathcal{C}_y$ is generated.

15. The method of claim 13 further including the step of identifying all non-empty subsets for each set $\mathcal{C}_y$.

16. The method of claim 8 wherein a partition is complete and nonoverlapping if $\mathcal{P}(\mathcal{Y}) = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_m\}$ satisfies $\bigcup_{i=1}^{m} \mathcal{G}_i = \mathcal{Y}$ and $\mathcal{G}_j \bigcap \mathcal{G}_k = \phi$ for any $j \neq k$, where each $\mathcal{G}_i \in \mathcal{P}(\mathcal{Y})$ is a group for $p(x,y)$, and $\mathcal{G}_j \cup \mathcal{G}_k$ and $\mathcal{G}_j \cap \mathcal{G}_k$ refer to the union and intersection respectively of the members of $\mathcal{G}_j$ and $\mathcal{G}_k$.

17. The method of claim 8 wherein said coding scheme is a lossless coding scheme.

18. The method of claim 8 wherein said coding scheme is a near-lossless coding scheme.

19. The method of claim 8 wherein said coding scheme is a side-information, lossless coding scheme.

20. The method of claim 8 wherein said coding scheme is a side-information, near-lossless coding scheme.

21. A method of code for X and Y comprising:

generating a partition pair $\mathcal{P}(\mathcal{X})$ and $\mathcal{P}(\mathcal{Y})$ such that each partition is a legitimate partition for a side-information, lossless decoding scheme;

identifying said partition pair as a legitimate partition for general lossless decoding if the two descriptions together give enough information to decode X and Y uniquely.

22. The method of claim 21 wherein said partition pair is a legitimate partition pair when for any $x, x' \in \mathcal{X}$ such that $\{\gamma_X(x), \gamma_X(x')\}$ does not satisfy the prefix condition, $\{\gamma_Y(y) : y \in \mathcal{A}_x \cup \mathcal{A}_{x'}\}$ satisfies the prefix condition.

23. The method of claim 21 wherein said partition pair is a legitimate partition pair when for any $y, y' \in \mathcal{Y}$ such that $\{\gamma_Y(y), \gamma_Y(y')\}$ does not satisfy the prefix condition, $\{\gamma_X(x) : x \in \mathcal{B}_y \cup \mathcal{B}_{y'}\}$ satisfies the prefix condition.

24. A method for generating a MASC code comprising:

generating instantaneous code by:

generating subtrees $\mathcal{T}_\mathcal{X}$ and $\mathcal{T}_\mathcal{Y}$ descending from nodes $\mathbf{n}_\mathcal{X}$ and $\mathbf{n}_\mathcal{Y}$ (including $\mathbf{n}_\mathcal{X}$ and $\mathbf{n}_\mathcal{Y}$ respectively).

25. The method of claim 24 further comprising satisfying one of the following conditions;

(A) $X \in \mathcal{T}_\mathcal{X}$ or $\mathbf{n}_\mathcal{Y}$ is a leaf implies that $Y \in \mathbf{n}_\mathcal{Y}$, and $Y \in \mathcal{T}_\mathcal{Y}$ or $\mathbf{n}_\mathcal{X}$ is a leaf implies that $X \in \mathbf{n}_\mathcal{X}$;

(B) $X \in \mathcal{T}_\mathcal{X}$ implies that $Y \notin \mathbf{n}_\mathcal{Y}$;

(C) $Y \in \mathcal{T}_\mathcal{Y}$ implies that $X \notin \mathbf{n}_\mathcal{X}$.

26. The method of claim 25 wherein said instantaneous code is lossless when:

generating code such that for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with $p(x, y) > 0$, final nodes $(\mathbf{n}_\mathcal{X}, \mathbf{n}_\mathcal{Y})$ are generated that satisfy;

(D) $(x, y) \in \mathbf{n}_\mathcal{X} \times \mathbf{n}_\mathcal{Y}$ and for any other $x' \in \mathbf{n}_\mathcal{X}$ and $y' \in \mathbf{n}_\mathcal{Y}$, $p(x, y') = p(x', y) = p(x', y') = 0$

27. A method of generating code comprising:

obtaining an alphabet of symbols generated by a data source

determining which of said symbols can have identical code descriptions and which symbols cannot have identical code descriptions;

28. The method of claim 27 further including determining which of said symbols can have code descriptions for which one symbols's code description is a prefix of another symbol's code description.

29. A method of generating code for data sources X and Y having data rates Rx and Ry respectively, comprising:

generating a code that minimizes $\lambda R_X + (1-\lambda) R_Y$ for an arbitrary value of λ.

30. The method of claim 29 wherein $\lambda \in [0,1]$.

31. A method for encoding and decoding a plurality of data streams comprising:

encoding said plurality of data streams using a plurality of encoders to produce a plurality of encoded data streams;

providing said plurality of encoded data streams to a receiver;

decoding said plurality of encoded data streams using a single decoder.

32. The method of claim 31 wherein said encoding and decoding are lossless.

33. The method of claim 31 wherein said encoding and decoding are near-lossless.

34. The method of claim 31 wherein said decoding is accomplished using side-information.

35. A method of designing codes comprising:

obtaining an alphabet of symbols generated by a data source;

ordering said alphabet of symbols;

identifying restrictions of a class of codes based on said ordering of said alphabet;

designing code for said restricted class for said ordering of said alphabet.

36. The method of claim 35 wherein said restrictions include a requirement that symbols be adjacent symbols.

37. The method of claim 35 further including the step of selecting an ordering of said alphabet based on generating code for a plurality of orderings.

38. The method of claim 37 wherein an ordering is selected based on a best rate resulting from one of said orderings.